



GRADUATION PROJECT 2024/2025

Faculty of Engineering
AL-AZHAR University

LogiDrive AI Vision Pro

➤ Proposal of SMART VEHICLE USING Logger,
FOTA and AI

- Omar Mohamed Youssef	 01114122502		
- Mohamed Refat Hamdallah	 01128920159		
- Ammar Ashraf Mohamed	 01010922448		
- Noura Mohamed Gamal	 01060671956		
- Sara Gomaa Mahmoud	 01095247510		
- Mariam Ahmed Madany	 01024703357		
- Mennatullah Elamir Hassan	 01069500076		
- Manar Abdelraouf Mohamed	 01122656273		
- Farha Ahmed Mohammed	 01158658065		
- Moawia Mohammed Sabour	 01126741067		

TABLE OF CONTENTS

1.Motivation	3
2.Project Overview	4
3.Methodology	5
5.Main System	14
4.Hardware Architecture	16
6.Testing	16
7.Timeline	17
8.References	17

1. Motivation

Modern vehicles are increasingly reliant on advanced electronic systems to enhance safety, comfort, and performance. However, as the complexity of these systems grows, so does the risk of critical system failures, which may occur without prior notice to the driver. In many cases, drivers depend heavily on these systems for safe and efficient driving, unaware of any malfunctions until it is too late. This can lead to dangerous situations, costly repairs, and a poor user experience.

2010-2020: Over 100 million vehicle recalls were issued in the United States alone, with a significant portion attributed to software-related issues.

2018: One report noted that nearly 50% of automotive recalls that year were due to software defects.

2019: Software issues were the cause of around 15% of all automotive recalls in the U.S., affecting millions of vehicles.



To address this challenge, our project focuses on developing a smart car equipped with a logger system that monitors and verifies the functionality of all critical vehicle systems before the driver begins their journey. The logger ensures that any potential issues are detected and communicated to the driver in real-time, reducing the risk of unexpected system failures during driving. Additionally, the car is integrated with FOTA (Firmware Over-The-Air) technology, allowing for remote software updates and real-time communication with the service center, further enhancing the vehicle's reliability and performance.

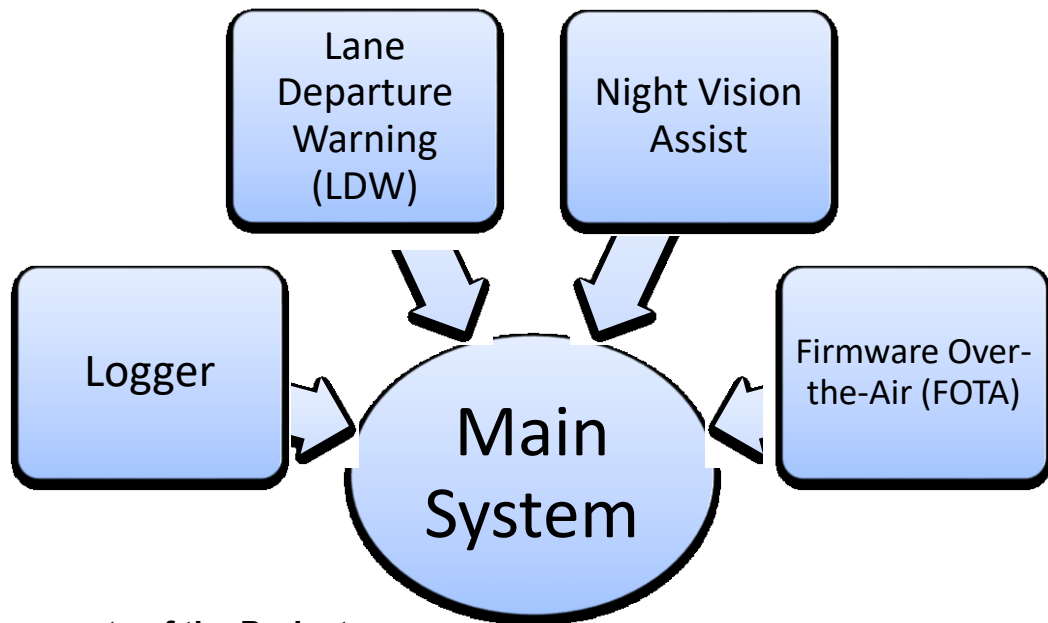
By keeping the driver informed of the vehicle's status at all times, our system aims to provide a safer and more reliable driving experience. This proactive approach not only prevents unexpected breakdowns but also ensures that the driver can make informed decisions, ultimately enhancing road safety and reducing maintenance costs.



2. Project Overview

In general, we will use one vehicle:

- The main vehicle has all the features.
- The project focuses on developing an intelligent vehicle system that leverages Advanced Driver Assistance Systems (ADAS) and Artificial Intelligence (AI) for real-time diagnostics, firmware updates, and failure monitoring.



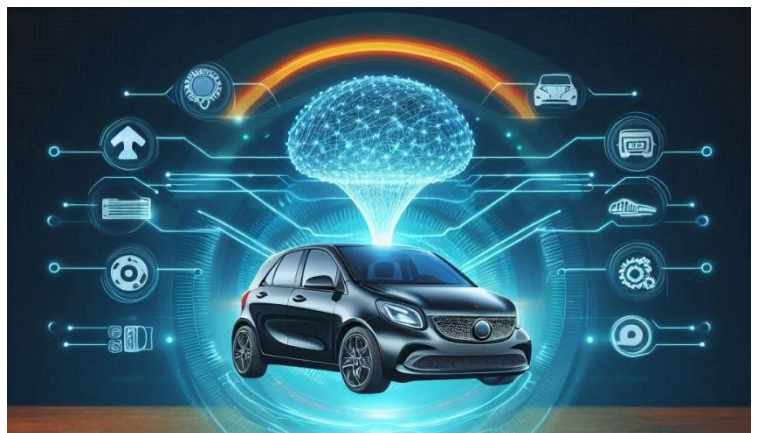
Key Components of the Project:

Logger System: A critical feature to track and log subsystem failures, ensuring that issues are identified and addressed proactively to avoid potential dangers on the road.

Firmware Over-The-Air (FOTA) Updates: This allows vehicles to receive software updates remotely, addressing software issues without the need for physical maintenance.

AI Vision: AI-based systems like Lane Departure Warning (LDW) and Night Vision ensure real-time decision-making to improve driver safety and comfort.

Hardware and Architecture: The project involves multiple microcontrollers, such as the STM32F10x ARM controller for subsystem management and Raspberry Pi for AI processing, particularly for vision-based systems like night vision.



3. Methodology

Firmware Over-the-Air (FOTA)

Firmware Over-the-Air (FOTA) is a technology that allows manufacturers to wirelessly upgrade and update device firmware during runtime without the need to connect directly to the device. Upgrades and updates are downloaded by FOTA-capable devices straight from the service provider.

FOTA is particularly useful when it comes to automotive embedded systems, especially those with large numbers of connected devices (ECUs) in the same car that need frequent updates.

In Automotive a car had at least 20 million to 30 million lines of code and 100-200 different ECUs. Each ECU has to be done manually, which was time-consuming and expensive, often even cost-prohibitive or outright impossible, before platforms started offering firmware over-the-air upgrading services.



FOTA mechanism:

Secure boot is divided into three sections:

1. Protecting the Boot process
2. Ensuring Integrity in software updates
3. Secure Delivery infrastructure

Protecting the booting process

It is a security feature designed to ensure that the system boots using only software that is trusted by the manufacturer, thus preventing the execution of malicious or unauthorized code during the system startup process.

Secure Boot is done by the use of digital signatures and certificates to verify the authenticity of software.

The verification process ensures that:

1. The software has not been modified
2. The software is from a trusted source

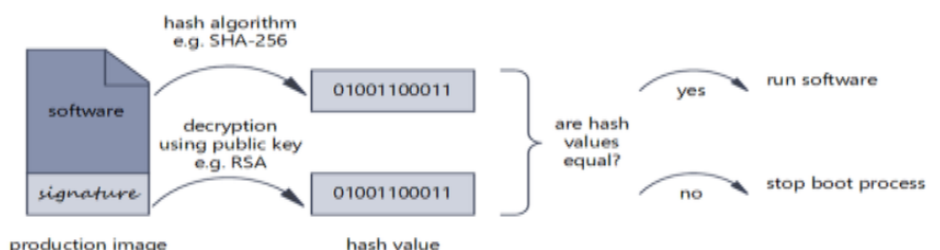
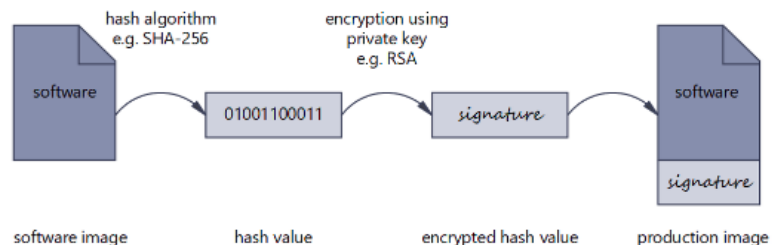
Ensuring Software Integrity

When delivering over-the-air (OTA) or manual updates for firmware or software,

it's essential to ensure that the update is complete, authentic, and has not been tampered with.

A mix of **cryptographic signatures** and **Hashing Algorithms** are used to ensure that only valid, unmodified updates are applied to the system.

The **bootloader** plays a critical role in the secure update process. It verifies the digital signature and integrity of the update before allowing it to be installed a **rollback mechanism** may be implemented in the bootloader, where the previous firmware version is kept in memory in case the new update fails. This way, if the update is incomplete or fails during installation, the system can roll back to the previous working version and remain operational.



Secure Delivery Infrastructure

To ensure that only authorized devices can access the update server, the system must implement strong authentication and authorization mechanisms like Mutual Authentication.

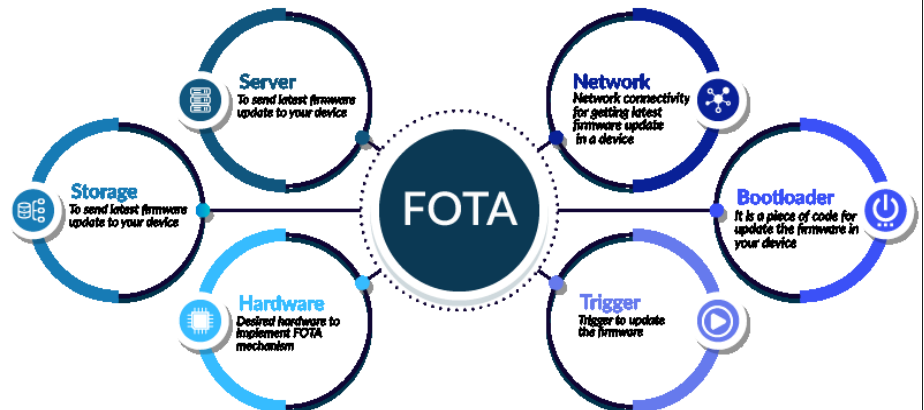
Also the update server itself must be secure to prevent attackers from uploading malicious updates or gaining access to sensitive data.

FOTA enables updates to:

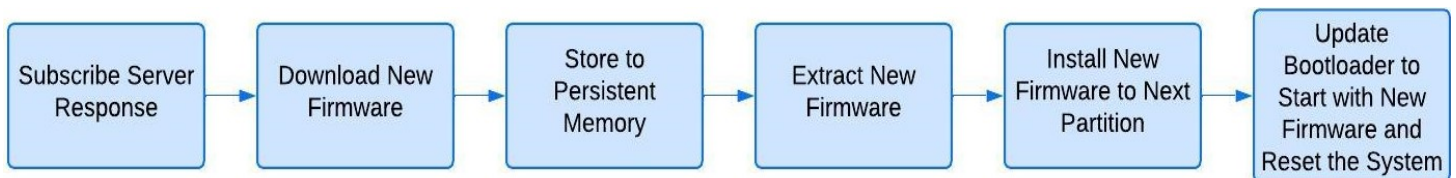
- Fix bugs or security vulnerabilities.
- Add new features or enhance existing ones.
- Improve performance.

Software

- **RTOS:** If the vehicle starts to speed and current location to take reaction at the right time.
- **ARM(Control Algorithms):** Determine if and how much another vehicle is closed to my vehicle
- **Feedback Systems:** Generate alerts or notifications to warn the driver there is update software detected will be not completed.
- **Integration Software:** Ensures smooth communication between cloud and other vehicle's main systems like LDW, and Night Vision to be updated.

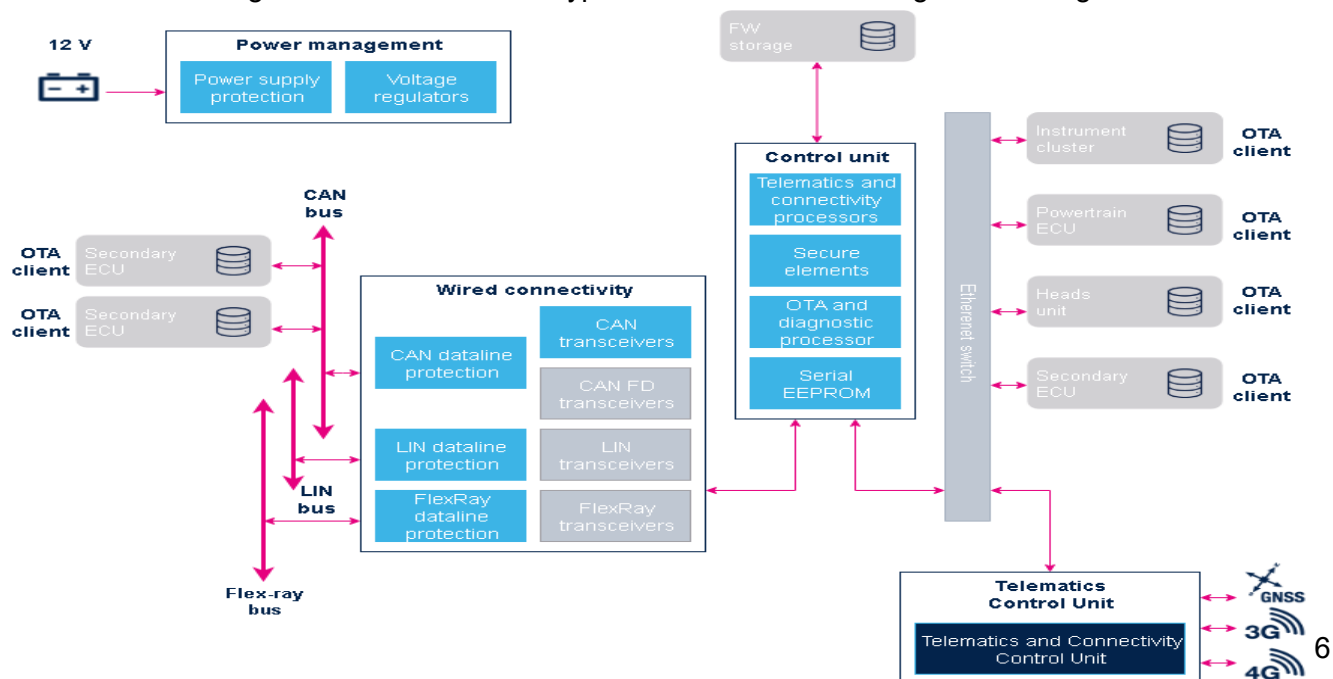


System Flow Chart:



Hardware

- **Telematics unit (ESP8266):** used for wireless communication (Wi-Fi) between the vehicle and cloud.
- **Microcontroller (STM32F103C8T6):** is the central unit that controls and manages all operations within each vehicle.
- **Main ECU:** Will receive the update from telematics unit and important data about the update like the target ECU ID and will decrypt the code before sending it to the target ECU.



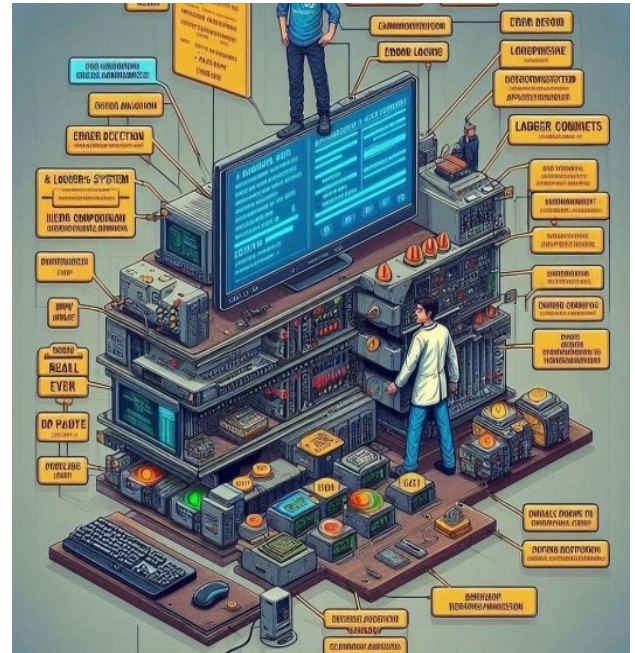
Logger System

Logger Overview

The logger is a crucial component of the ADAS (Advanced Driver Assistance System) in the smart car architecture. It operates as a separate sub-system using a Cortex-M ARM controller, responsible for monitoring, detecting, and reporting any failures in the various ADAS subsystems, such as Lane Departure Warning (LDW) and Night Vision System.

Logger Responsibilities

1. **Monitor Sub-system Failures:** The logger continuously monitors the health of all ADAS subsystems (ex: LDW, NVS) to detect malfunctions, anomalies, or failures in real-time.
2. **Report to Main System:** Upon detecting any subsystem failure, the logger immediately sends an error report to the main system for display on the dashboard and to notify the driver.
3. **Log Data:** The logger saves detailed information about subsystem failures in its internal database. This data can be used for future diagnostics, analysis, and trend tracking to improve system reliability.
4. **Upload Data to Cloud:** The logger can periodically upload logged data to a secure cloud storage or diagnostic server. This facilitates remote monitoring, fleet management, or advanced data analytics, enabling over-the-air (OTA) diagnostics and updates to enhance system performance.



Subsystem Responsibilities

- ## 1.LDW (Lane Departure Warning) Diagnostic Responsibilities:
- Monitor lane detection algorithms for failures.
 - Ensure proper functioning of steering actuators and log any calibration or actuator errors.
- ## 2.Vision Diagnostic Responsibilities:
- Monitor camera and sensors for proper functionality.
 - Detect processing delays or inaccuracies screen.

Diagnostics Mechanism

This work can be divided into several steps, including Self-Diagnostic for each ADAS system, Error Logging, and Integration with FOTA. Below is a detailed outline of how to implement each step.

1. Self-Diagnostic for Each ADAS System:

- **Concept:** Each system within ADAS (such as cameras, sensors, and processing units) will need to perform regular checks to detect any faults. This can be achieved by programming a Diagnostic Module within each system.

- Key Components of Diagnostics:

- **Health Checks:** Each sensor or camera will perform self-assessments (e.g., ensuring data accuracy, signal clarity, and checking for deviations or damages).
- **Inter-System Communication:** In case of a fault, a report containing an Error Code and Fault Details should be sent to the main control unit or logger.

2. Error Management (Logger System)

- **Receiving Reports:** The logger will collect reports from all systems performing self-diagnostics. These reports will contain:

- **Error Code:** Indicating the type of issue (e.g., camera failure, connection failure, etc.).
- **Additional Details:** Such as the time of the issue, the affected system, etc.
- **Error Analysis:** The logger can filter or prioritize critical errors before sending them to a higher system (such as FOTA).

3. Integration with FOTA System

- **Notification Generation:** When the logger detects an error, it generates a Notification or Report and raises it to the FOTA system. The report will include:

- **Problem Code:** A standard code that can be interpreted remotely.
- **Specific Fault Location:** Such as the damaged sensor or camera.
- **Other Details:** Additional information such as recommended repair steps or system status at the time of the fault.

- **Remote Updates:** If the issue requires a fix, the FOTA system can send firmware updates to correct errors in the affected systems.

4. Technical Implementation

-In Each System:

- Utilize Watchdog Timers and Sensors Health Monitoring to perform self-diagnostics periodically.
- For each potential fault, program a unique error code that is sent upon detection.

-In Logger:

- Program a unit to receive diagnostic messages using CAN protocol
- Build a system to process error codes and generate reports using a standardized format.

-Integration with FOTA:

- Design an interface between the logger and the FOTA system using a suitable protocol (e.g., HTTP, MQTT, or other wireless protocols).

Diagnostics Software Implementation :

The main objectives of this is to Implement the UDS protocol by Focusing on essential UDS services for example:

- **Service 0x19:** Diagnostic Trouble Codes (DTC)
 - Reading DTCs
 - Clearing DTCs
- **Service 0x22:** Data Identifier (DID)
 - Reading Data by Identifier
- **Service 0x31:** Routine Control
 - Executing specific diagnostic routines.

➤ Reasons for Considering the Development of a Simple UDS System:

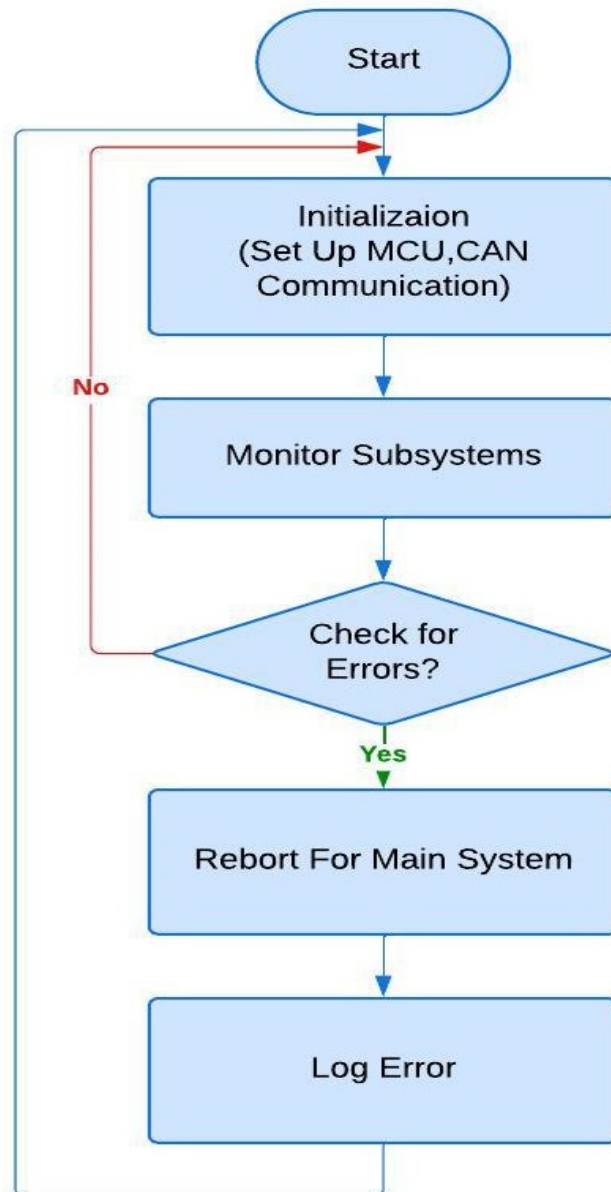
1. Cost Control:

Licensing fees for specialized diagnostic software, such as UDS, can be extremely high, often exceeding the budgets of graduation projects for students and by developing a custom UDS system, we can avoid these exorbitant licensing costs.

2. Customization According to Needs:

Developing a tailored UDS system enables us to customize functions based on our project requirements. This flexibility allows us to develop solutions that meet specific needs, such as reading and clearing diagnostic trouble codes (DTCs), without relying on unnecessary features.

System Flow Chart:



ADAS Systems:

To Implement and apply the logger system and achieve integration with the Main System, we chose Two of ADAS Systems:

1. Vision System powered by AI
2. Lane Departure Warning (LDW)

Let's elaborate on them.

Vision Assistance

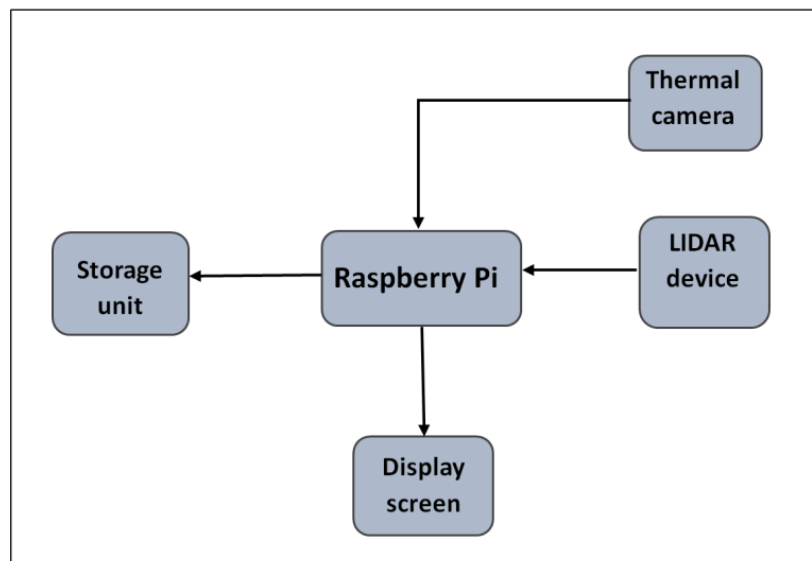
At night, there is 60 percent less traffic on the roads. Despite this, 50 percent of fatal accidents happen after dark, as our ability to perceive distance and detect objects is severely impaired at night. Now with the combined power of LIDAR and AI, night vision has become more accurate and safer, Reducing the risk of accidents and enhancing driving safety in the dark.

The Night Vision System is designed to provide enhanced visibility in low-light or completely dark conditions. The system captures images using thermal cameras or infrared cameras and uses LIDAR to obtain 3D distance and object data. These images are processed using computer vision and artificial intelligence techniques to improve clarity and detect objects. The enhanced images are then displayed on a screen for the driver.



Hardware:

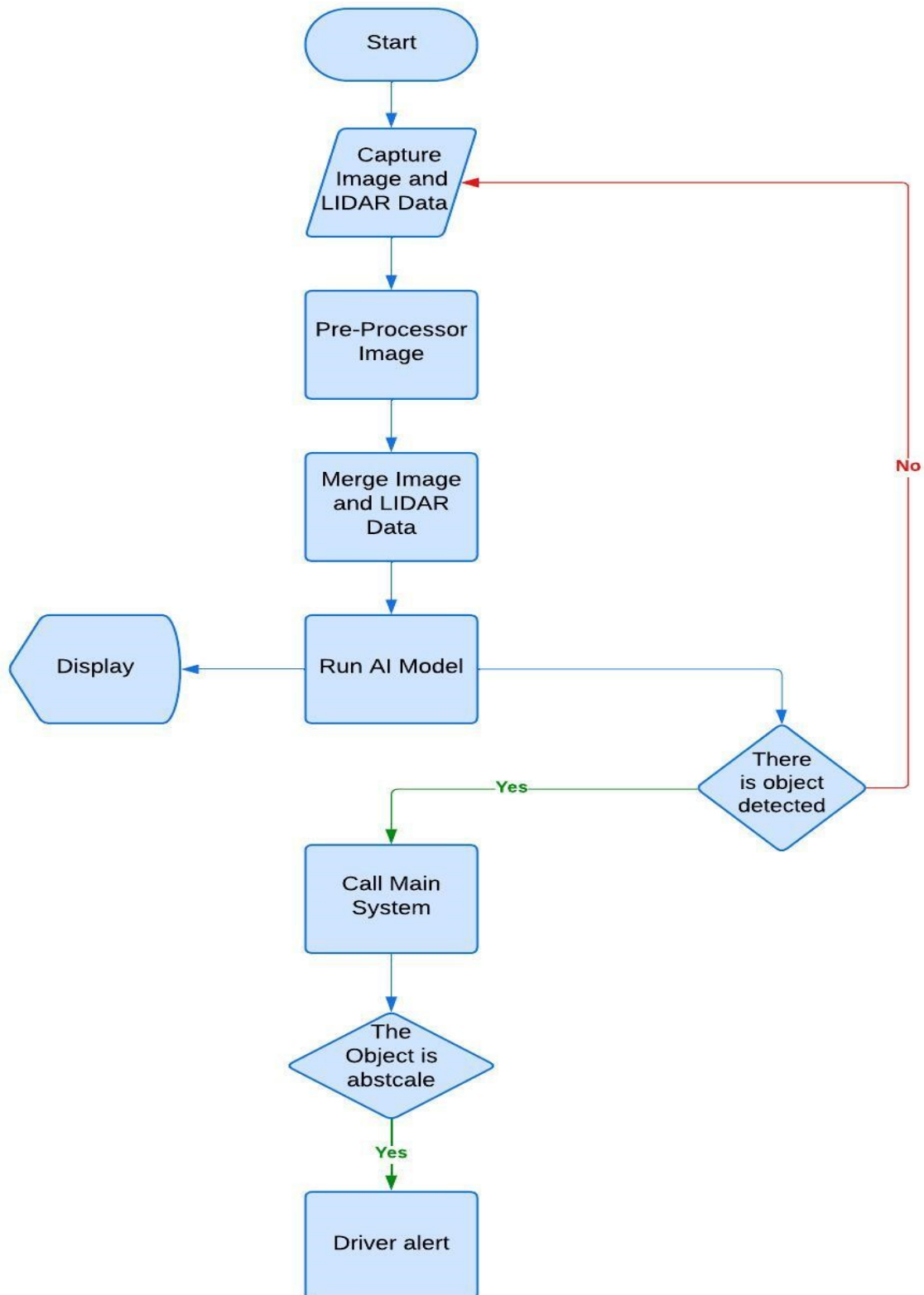
- **Thermal Camera or Infrared Camera:** For capturing images in the dark.
- **LIDAR Device:** For obtaining 3D distance and object data.
- **Raspberry Pi:** For processing images and



Software:

- **Raspberry Pi Operating System:** Such as Raspberry Pi OS.
- **OpenCV:** For image processing.
- **Tensor Flow:** For running AI models.
- **Python:** For writing the software

System Flow Chart:



Lane Departure Warning (LDW)

LDW is a critical feature in modern vehicles designed to enhance safety by helping drivers stay within their lanes.

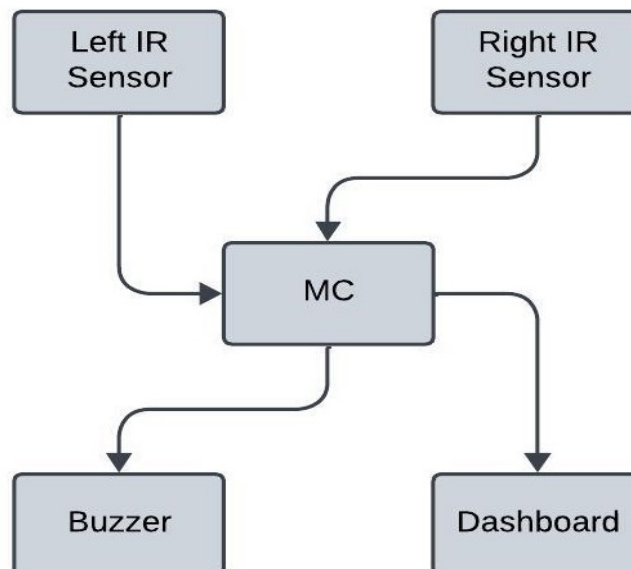
Using a combination line follower mounted on the vehicle, LDW monitors lane markings on the road. If the system detects unintended lane departure without signaling, it provides visual or auditory alerts to the driver.

Some systems also actively intervene by gently steering the vehicle back into its lane. This technology helps prevent accidents caused by drifting and promotes safer, more focused driving



Hardware

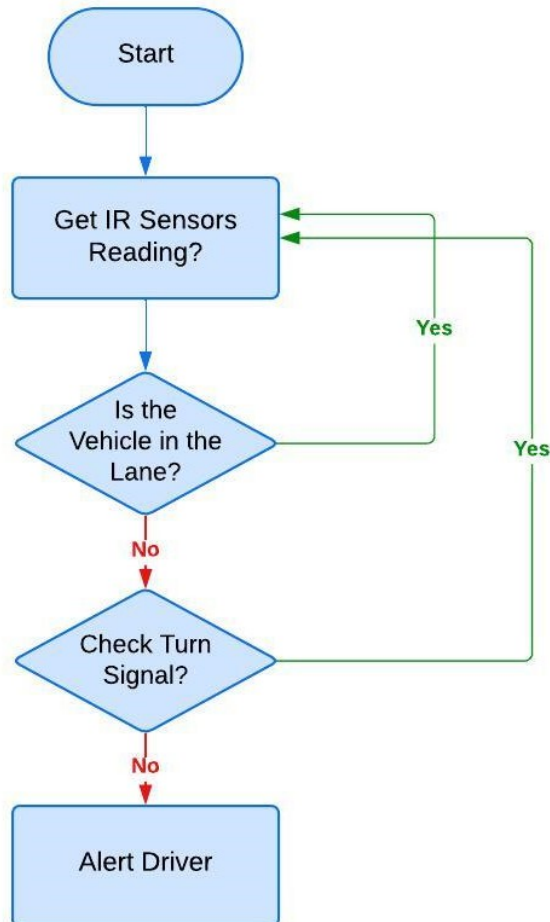
- **Sensors:** line follower, they help detect lane boundaries and vehicle position.
- **Alert:** If the vehicle starts to drift out of its lane start turning on the alarm.
- **Steering Actuators:** Mechanisms that adjust steering based on system commands.
- **Display (Monitor/Screen):** provides visual feedback to the driver, showing lane markings, system status, and alerts when the vehicle drifts out of its lane.



Software

- **RTOS:** If the vehicle starts to drift out of its lane start tuning on the alarm or take reaction at the right time.
- **ARM: Control Algorithms:** Determine if and how much steering adjustment is needed to keep the vehicle within its lane.
- **Feedback Systems:** Generate alerts or notifications to warn the driver if lane departure is detected.
- **Integration Software:** Ensures smooth communication between LDW and other vehicle systems like steering and braking

System Flow Chart



Main System

Main System Responsibilities and Interactions:

Key Responsibilities:

1. Subsystem Management:

- **Control:** The main system is responsible for turning ADAS subsystems on or off as needed. This includes managing the operational status of systems such as Lane Departure Warning (LDW) and night vision system.
- **Integration:** It ensures that all subsystems are correctly integrated and functioning together to provide a cohesive driving experience.

2. Vehicle Control:

- **Steering:** The main system directly controls the steering mechanisms of the vehicle, ensuring precise and responsive handling.
- **Braking:** It also manages braking functions, coordinating with other systems to apply brakes as required.

3. Monitoring:

- **Vehicle Status:** Continuously monitors the car's status, including speed and the operational state of various subsystems.
- **Subsystem Activity:** Keeps track of which subsystems are active and their current states.

4. Alert Management:

- **User Notifications:** Sends alerts to the driver through visual indicators on the monitor or audible sound alarms. Alerts may be triggered by system statuses or detected issues.

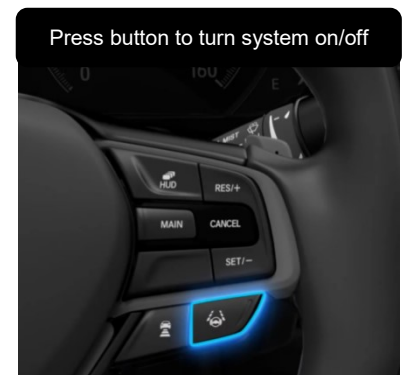
Subsystem Interactions:

1. Lane Departure Warning (LDW):

- **Lane Correction:** Similarly, if LDW requires adjustments to keep the vehicle within its lane, it sends a request to the main system to take corrective action.

2. Vision System:

- **Object Detection:** The vision subsystem detects objects in the vehicle's path. If an object is detected, it sends a message to the main system.
- **Alerting:** If the detected object is a potential hazard, the main system will issue an alert to the driver.
- **Object Handling:** While the detected object is in lanes, the main system will delegate control to the LDW to manage the situation.



Communication:

- **CAN Bus:** Communication between the main system and subsystems is handled through the Controller Area Network (CAN). This robust communication protocol ensures reliable data exchange and coordination among all system components.

Additional Tasks for the Main System

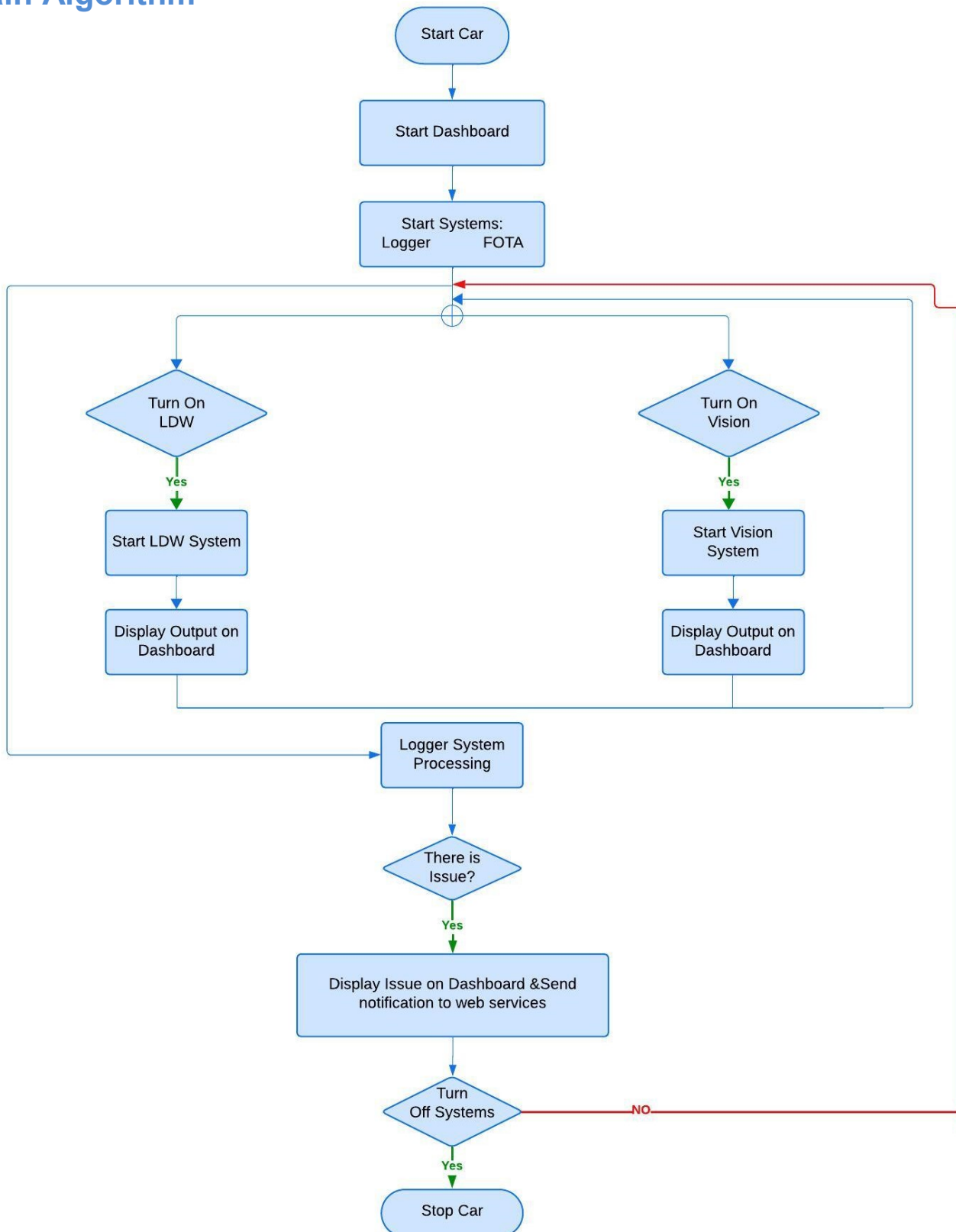
1. System Health Monitoring:

- **Diagnostics:** Monitors the health and status of each subsystem to ensure proper operation. Detects faults or malfunctions and can trigger alerts or initiate fail-safe measures if needed.
- **Self-Test:** Periodically performs self-diagnostics to ensure the main system and connected subsystems are functioning correctly.

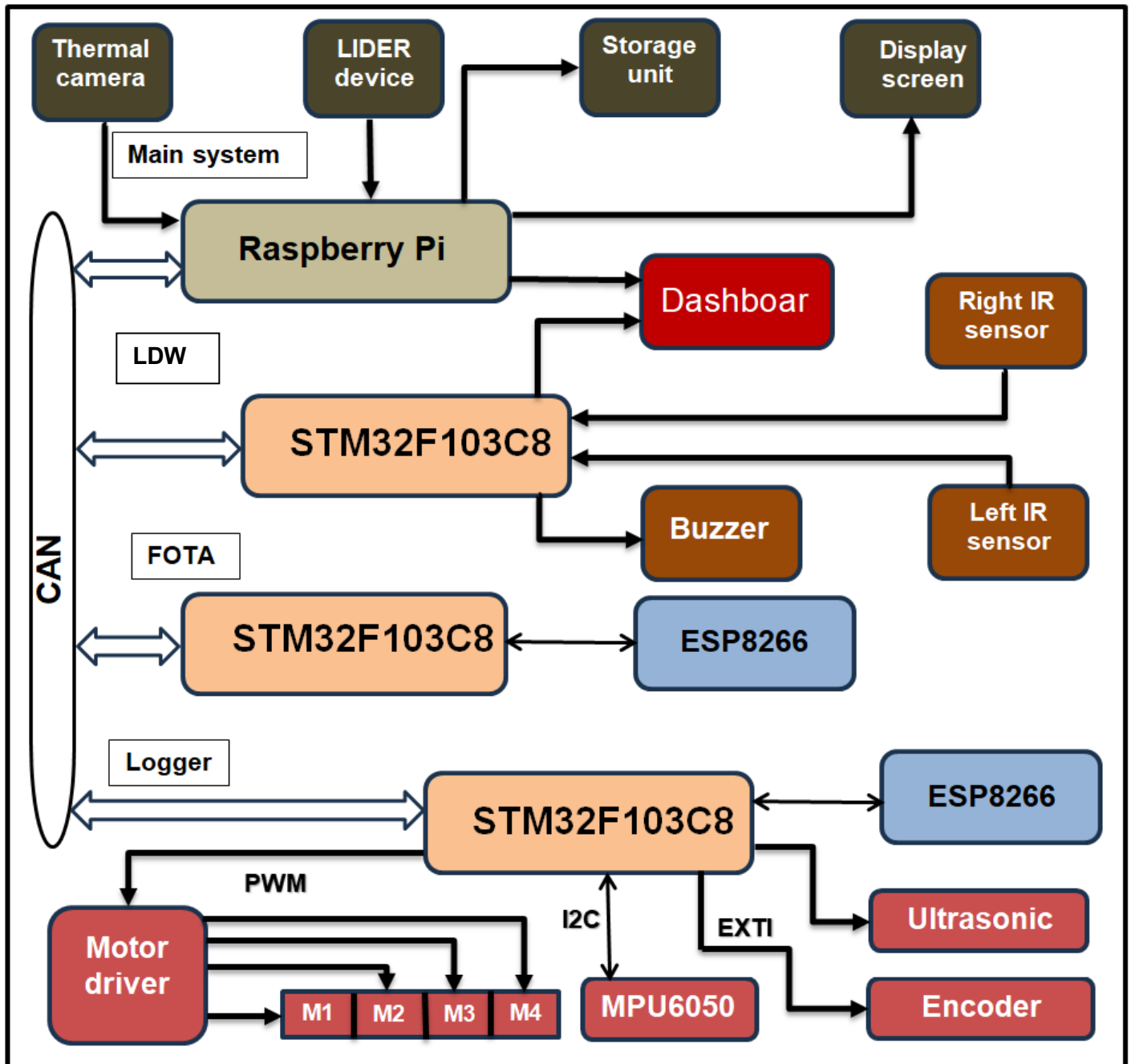
2. User Interface and Interaction:

- **Driver Feedback:** Provides feedback to the driver through visual or auditory signals about the status of the vehicle and any active alerts or warnings.
- **Custom Settings:** Allows drivers to customize settings for various subsystems according to their preferences.

Main Algorithm



4. Hardware Architecture



5. Testing

CARLA simulator

What is CARLA Simulator?

CARLA (CAR Learning to Act) is an open-source simulator for autonomous driving research. Developed by the Computer Vision Center in Barcelona, Spain, CARLA provides a platform for testing and developing autonomous driving systems in realistic urban environments.



Using CARLA as a simulation platform can significantly enhance the development and testing of the Advanced Driver Assistance Systems (ADAS) subsystems described in your document. Here's how CARLA can improve each of the subsystems:

1. Lane Departure Warning (LDW)

- **Simulating Diverse Road Conditions:** CARLA allows us to simulate various road environments with different lane markings, curves, and intersections. We can test how the LDW system performs in different scenarios, such as sharp curves, faded lane markings, or complex intersections.
- **Testing under Different Weather and Lighting Conditions:** The ability to change weather conditions and time of day in CARLA helps us evaluate how the LDW system behaves under rain, fog, or low-light conditions. This is crucial for ensuring the robustness of the LDW system.

2. Vision

- **Low-Light and Night-Time Scenarios:** CARLA can simulate night driving scenarios where the vision system can be tested for detecting pedestrians, animals, or obstacles that might not be visible with regular headlights. You can adjust the lighting conditions to represent urban and rural night environments.
- **Sensor Fusion Testing:** CARLA supports multiple sensor types, allowing you to test how the vision system integrates with other sensors like LIDAR to enhance detection accuracy in low visibility.

So by leveraging CARLA, you can enhance the reliability, safety, and effectiveness of your ADAS subsystems, ensuring they perform well in a wide range of real-world conditions.



6. Timeline



7. References

- https://www.researchgate.net/figure/Architecture-of-proposed-network_fig1_358691941
- <https://www.bankrate.com/insurance/car/car-recall-facts-and-statistics/>